TEL AVIV UNIVERSITY
Department of Computer Science
0368.4281 – Advanced topics in algorithms
Spring Semester, 2011/2012

Homework 3, May 19, 2012

Due on Sunday June 3. Please keep a copy of the homework and put a hardcopy in my mailbox if you can !

1. A semi-splaying is a variation of splaying where in the zig-zig step we only do the top rotation (on the edge (y, z) in the slides) and continue from y. Prove that the access lemma holds for semi-splaying.

2. Consider the following top-down splaying operation. We start with the root as the current node. From the current node, we look ahead two steps (unless we reach the target node in one step in which case we stop). If these two steps are in the same direction (i.e. left-left or right-right), rotate the top edge and continue from the original grandchild. If they are in opposite directions, look ahead one more step (to the great-grandchild of the current node, the child of x in the slides which is either in subtree B or in subtree C). As in zig-zag we rotate at the second edge down and then at the top edge, and continue from the original great-grandchild. (This is the zig-zag case with one step not corresponding to a rotation.) Prove that the access lemma holds for this version of splaying.

3. Show how to extend dynamic trees to support the operation $evert(v)$. This operation changes the actual tree containing v such that following the operations v is the root of this tree. The directions of all the edges from v to the previous root of the tree are reversed. An efficient implementation should run in $O(\log n)$ time. Do not hurt the logarithmic running time of any of the other operations.

4. On a set of n nodes we perform a sequence of operations, each of which is one of the followings.
(a) $insert(u, v)$: Adds an edge between u and v.
(b) $delete - oldest$: Removes the edge that was inserted first among the edges currently in the graph.
(c) $connected(u, v)$: Answers true if there is a path from u to v in the current graph. Otherwise answers false.

Describe a data structure that supports these operations and analyze its performance. (A structure supporting each operation in $O(\log n)$ amortized time would receive maximum score.)

5. Consider the following alternative (to FIFO, say) selection rule of active nodes in the push/relabel maximum flow algorithm. The algorithm performs passes over active nodes. In each pass it examines all the active nodes in nonincreasing order of the distance labels. While examining a node it discharges the node (i.e. it pushes flow from this node until either its excess becomes zero or it is relabeled.). In the next pass, it again examines active nodes in nonincreasing order of their new

distance label. If during a pass, the algorithm relabels no node then the algorithm stops.
a) Prove that this version of the push-relabel algorithm performs $O(n^3)$ non-saturating pushes.
b) Suggest an implementation of this version that runs in $O(n^3)$ time